What is claimed is:

1.   A method to process large volumes of data using many node hosts in a system comprises:

producing a directed graph of the programmable nodes that guides the flow of data and control of processing from one node to the next node through the system.

2.   The method of claim 1 further comprising:

forming a dynamic modification of the Data-Flow Map to automatically fail-over to redundant back-up nodes based on thresholds established for the component hosts.

3.   The method of claim 2 wherein the system provides 1 to N level redundancy of nodes.

4.   The method of claim 3 wherein each node includes a fault manager and node manager that share the data flow map and a subset of the data flow map.

5.   The method of claim 3 wherein the fault manager provides a back-up node for a primary node.

6.   The method of claim 4 wherein the fault manager that provides the backup node notifies all fault managers of other nodes by:

modifying the data flow map of the backup node.

7.   The method of claim 6 wherein the notification includes the node-ids of the primary node and the back-up node and the node-id includes the destination IP-Address of its node host and a listen-port.

8.   The method of claim 2 further comprising:

measuring information pertaining to operational status of a node by determining threads running in a node and processor resources provided to the node.

9.   The method of claim 8 further comprising:

periodically polling each node on a node host for the node's operational status condition.

10.   The method of claim 8 further comprising:

determining if a operational status measure of a node goes below a set threshold in order to notify other fault managers.

11.   The method of claim 8 further comprising:

determining the first available healthy back-up node; and

marking the first available back-up node as the primary node, while placing the primary node at the end of a list with an non-operational status.

12.   The method of claim 3 further comprising:

determining by a source node that a destination node cannot accept a transfer; and

indicating to the fault manager that the destination node is unhealthy to dynamically modify the data-flow map and re-direct the flow of data to a healthy back-up node for the unhealthy primary node.

13.   A computer program product residing on a computer readable medium for providing fault tolerance to processing nodes executing on host computers of a distributed network accounting system, comprises instructions for causing a computer to:

produce a directed graph of the programmable nodes that guides the flow of data and control of processing from one programmable node to the next programmable node through the system; and

form a dynamic modification of the data-flow map to automatically fail-over to redundant back-up programmable nodes based on thresholds established for the component hosts.

14.   The computer program product of claim 13 wherein the system provides 1 to N level redundancy of programmable nodes.

15.   The computer program product of claim 13 wherein computer program product executes as a fault manager in a processing domain that includes a node manager, that manages execution of the programmable nodes that share the data flow map and a subset of the data flow map.

16. The computer program product of claim 14 wherein the fault manager establishes a back-up node for a primary node.

17. The computer program product of claim 16 wherein instructions to cause the fault manager to establish the backup node further comprises instructions to notify all fault managers of other nodes by instructions to:

    modify the data flow map of the backup node.

18. The computer program product of claim 13 further comprising instructions to:

    measure information pertaining to health of a node by determining threads running in a node and processor resources provided to the node.

19. The computer program product of claim 13 further comprising instructions to:

    periodically poll each node on a node host for the node's health condition.

20. The computer program product of claim 15 further comprising instructions to:

    determine if a health measure of a node goes below a set threshold in order to notify other fault managers.

21. The computer program product of claim 13 further comprising instructions to:

    determine the first available healthy back-up node; and

    mark the first available back-up node as the primary node, while placing the primary node at the end of a list with an unhealthy status.

22. The computer program product of claim 15 further comprising instructions to:

    determine by a source node that a destination node cannot accept a transfer; and

    indicate to the fault manager that the destination node is unhealthy to dynamically modify the data-flow map and re-direct the flow of data to a healthy back-up node for the unhealthy primary node.

23.   A distributed network accounting system, comprising:

a plurality of host computers that host a network accounting system, and a computer program product residing on a computer readable medium for providing fault tolerance to a data processing domain of the network accounting system, comprises instructions for causing the host computer to:

produce a directed graph of the programmable nodes that guides the flow of data and control of processing from one programmable node to the next programmable node through the system; and

form a dynamic modification of the data-flow map to automatically fail-over to redundant back-up programmable nodes based on thresholds established for the component hosts.

24.   The system of claim 23 wherein computer program product executes as a fault manager in a processing domain that includes a node manager, that manages execution of the programmable nodes that share the data flow map and a subset of the data flow map.

25.   The system of claim 23 wherein the programmable nodes can be a data collector process that produces network accounting records, or an aggregation process that aggregates network accounting records, or an enhancement process that enhances attributes of network accounting records, or an output interface process that produces records for use by an application.

26   The system of claim 23 wherein the data processing domain further comprises:

a fault manager that executes the computer program to produce a dynamic modification of a directed graph.

27.   The system of claim 23 wherein the computer program products executes on a component that is a node manager, a local data manager, a remote data manager, an administrative server or an administrative client.

28.   The computer program product of claim 27 wherein the components that are nodes where changes in the processing context of the component are characterized as generally single/atomic transactions or other transactions the product further comprises instructions to:

context check-point a state of processing in the a data processing domain to permit automatic recovery of the data processing domain to the data processing domain's most recent processing context checkpoint; and

execute an operating system facility to provide the automatic recovery of the data processing domain to the data processing domain's most recent processing context.

29. The computer program product of claim 28 wherein the processing context of a component includes the entries in its configuration file or node manager table or global node-map.

30. A host computer for deployment in a distributed network accounting system, comprising:

a processor; and

a computer program product residing on a computer readable medium for providing fault tolerance to a network accounting process executed on the processor, comprises instructions for causing the processor to:

produce a directed graph of a group of programmable nodes that guides a flow of data and control of processing from one programmable node to the next programmable node through the distributed network accounting system; and

form a dynamic modification of the data-flow map to automatically fail-over to redundant back-up programmable nodes based on thresholds established for the component hosts.

31. The host computer of claim 31 wherein computer program product executes as a fault manager in the processing domain that includes a node manager, that manages execution of the programmable nodes that share the data flow map and a subset of the data flow map.

32. The host computer of claim 31 wherein the programmable nodes can be a data collector process that produces network accounting records, or an aggregation process that aggregates network accounting records, or an enhancement process that enhances attributes of network accounting records, or an output interface process that produces records for use by an application.

33 The host computer of claim 31 wherein the data processing domain further comprises:

a fault manager that executes the computer program to produce a dynamic modification of a directed graph.

34. A method for recovery of processing in a distributed network accounting system comprising many nodes executing on node hosts, the method comprises:

classifying nodes in the system according to complexity of processing in the node, and for nodes of relatively low processing complexity, context check-pointing a state of processing in the nodes to permit automatic recovery of the node to the nodes' most recent processing context checkpoint; and

for nodes of relatively high complexity, producing a directed graph of the programmable nodes that controls a flow of data and control of processing through the system, and producing a dynamic modification of the directed graph to automatically fail-over to redundant back-up nodes based on thresholds established for the component hosts.

35. The method of claim 34 wherein context check pointing provides 1 to 1 level of redundancy of node components.

36. The method of claim 34 wherein for nodes of relatively high complexity producing a directed graph provides 1 to N level redundancy of nodes.

37. The method of claim 34 wherein each component executes a recovery manager that provides fault tolerance of components by the context check pointing.

38. The method of claim 34 wherein the recovery manager uses operating system facilities to provide automatic recovery of the system components.